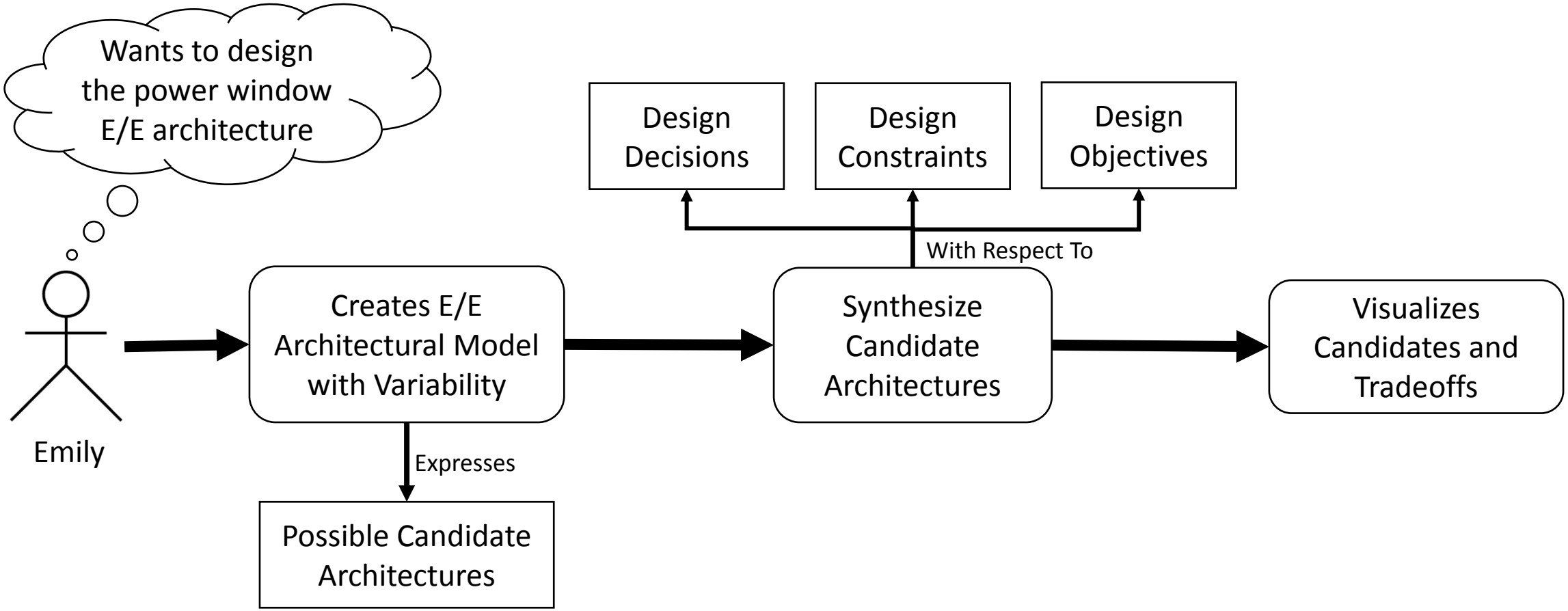
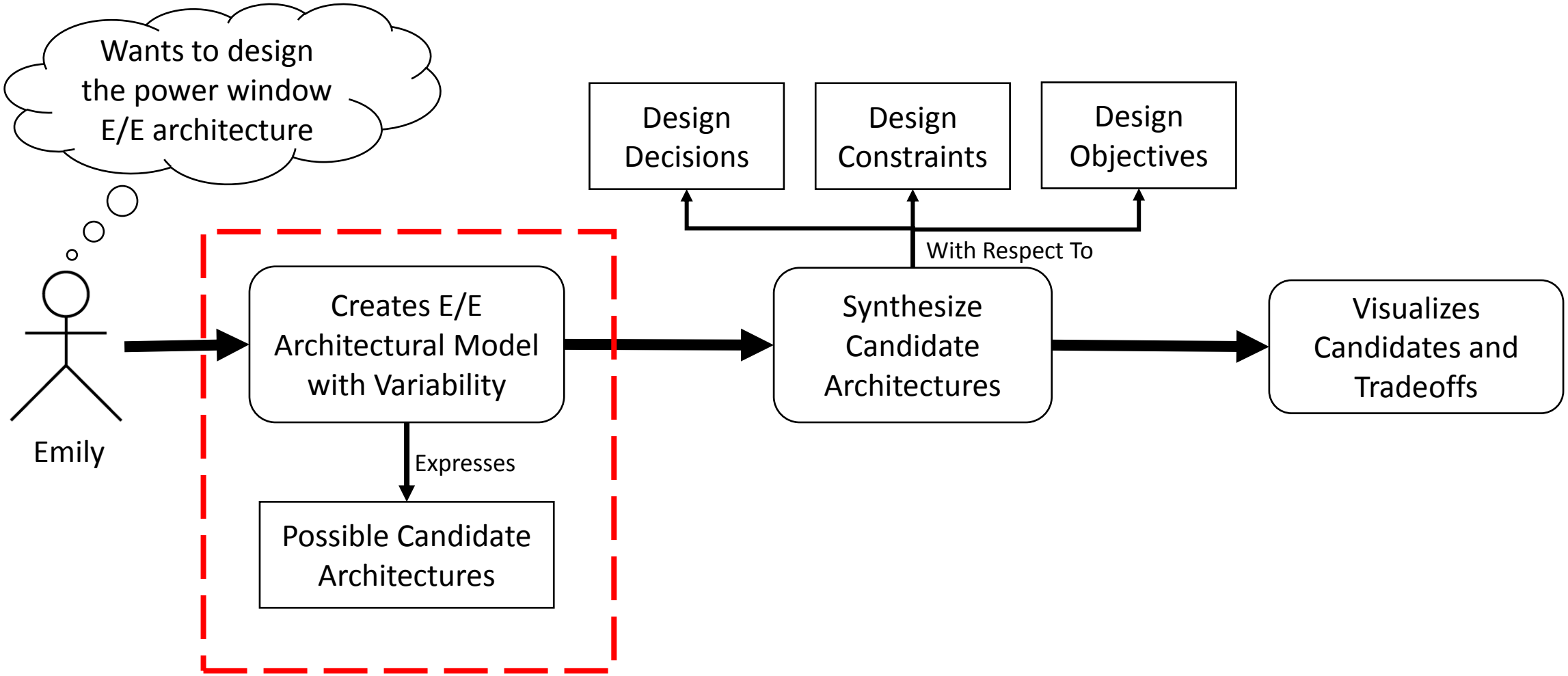
Synthesis and Exploration of Multi-Level, Multi-Perspective Architectures of Automotive Embedded Systems

Jordan Ross, Alexandr Murashkin, Jia Hui Liang, Michał Antkiewicz, Krzysztof Czarnecki

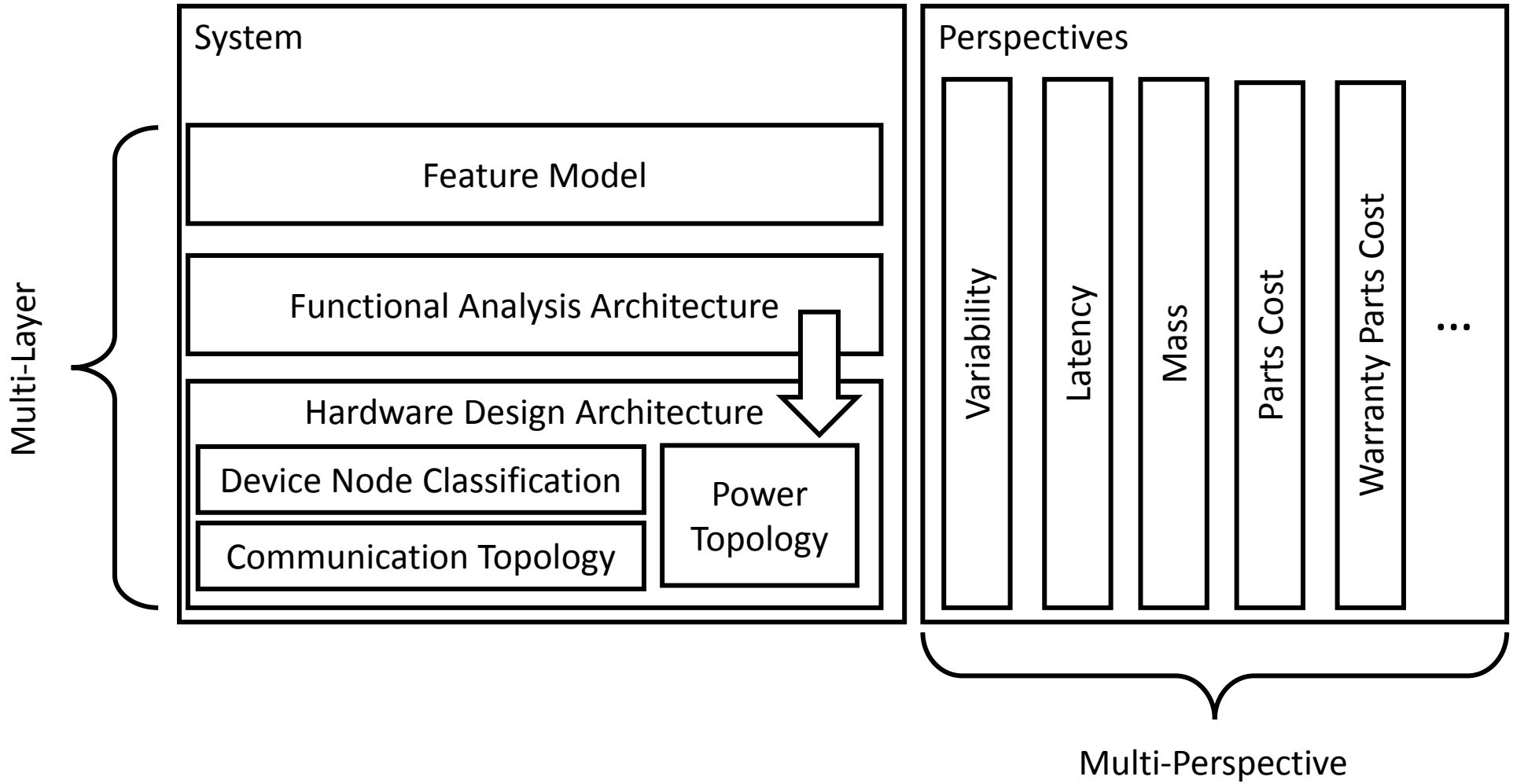
September 20th, 2017

Background & Motivation

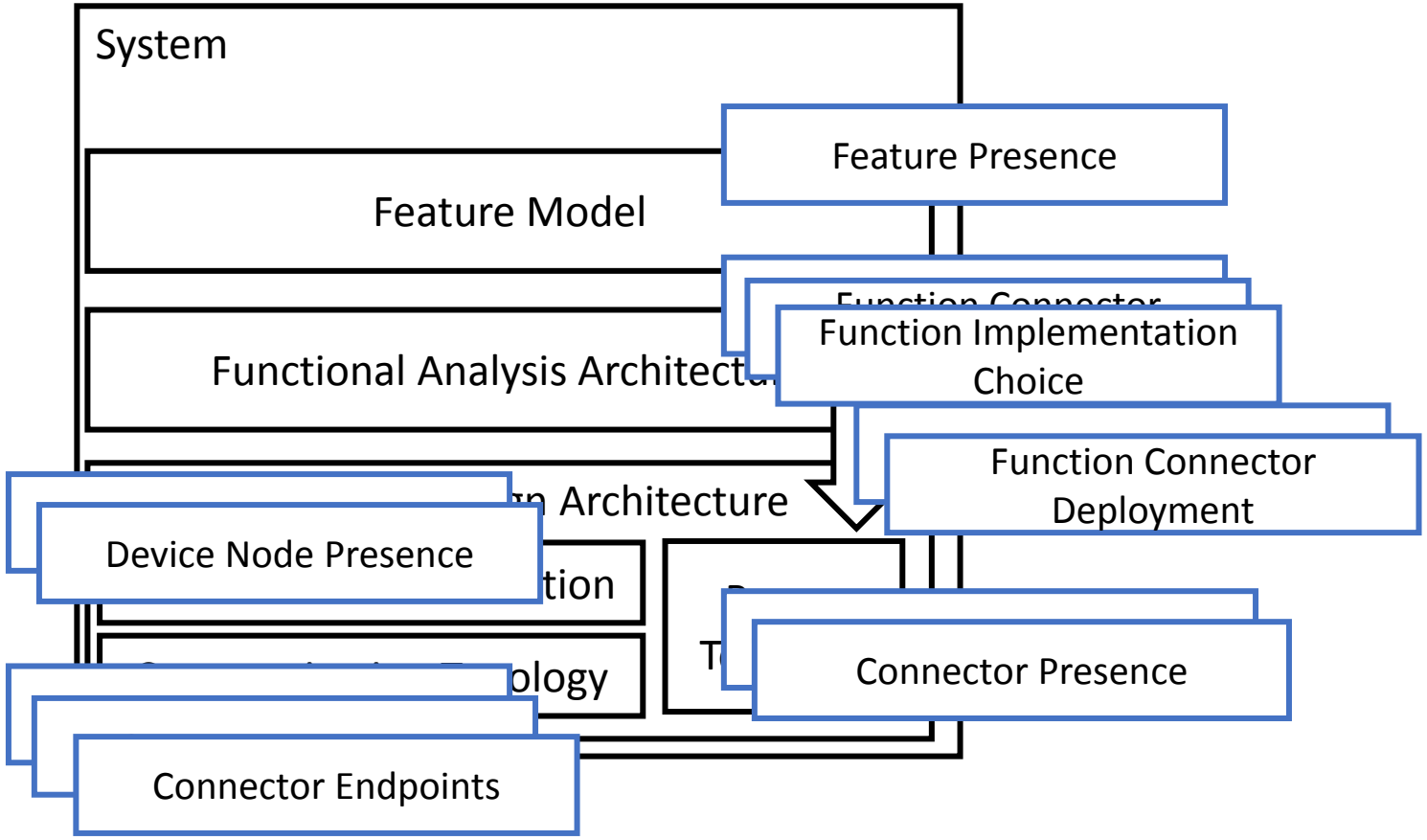



Wants to design the power window E/E architecture
Emily
Creates E/E Architectural Model with Variability
Expresses
Possible Candidate Architectures
Design Decisions
Design Constraints
Design Objectives
With Respect To
Synthesize Candidate Architectures
Visualizes Candidates and Tradeoffs

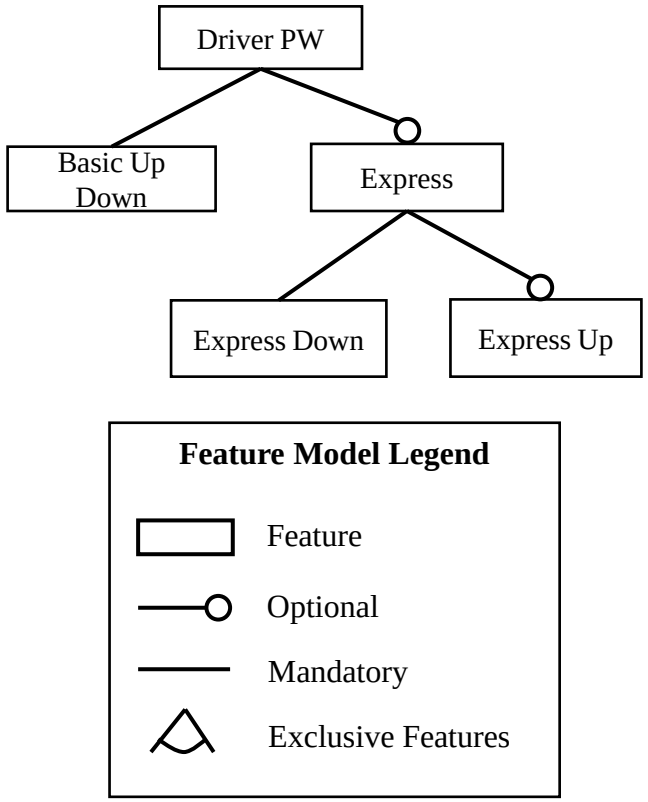
Our Reference Model for Early Design

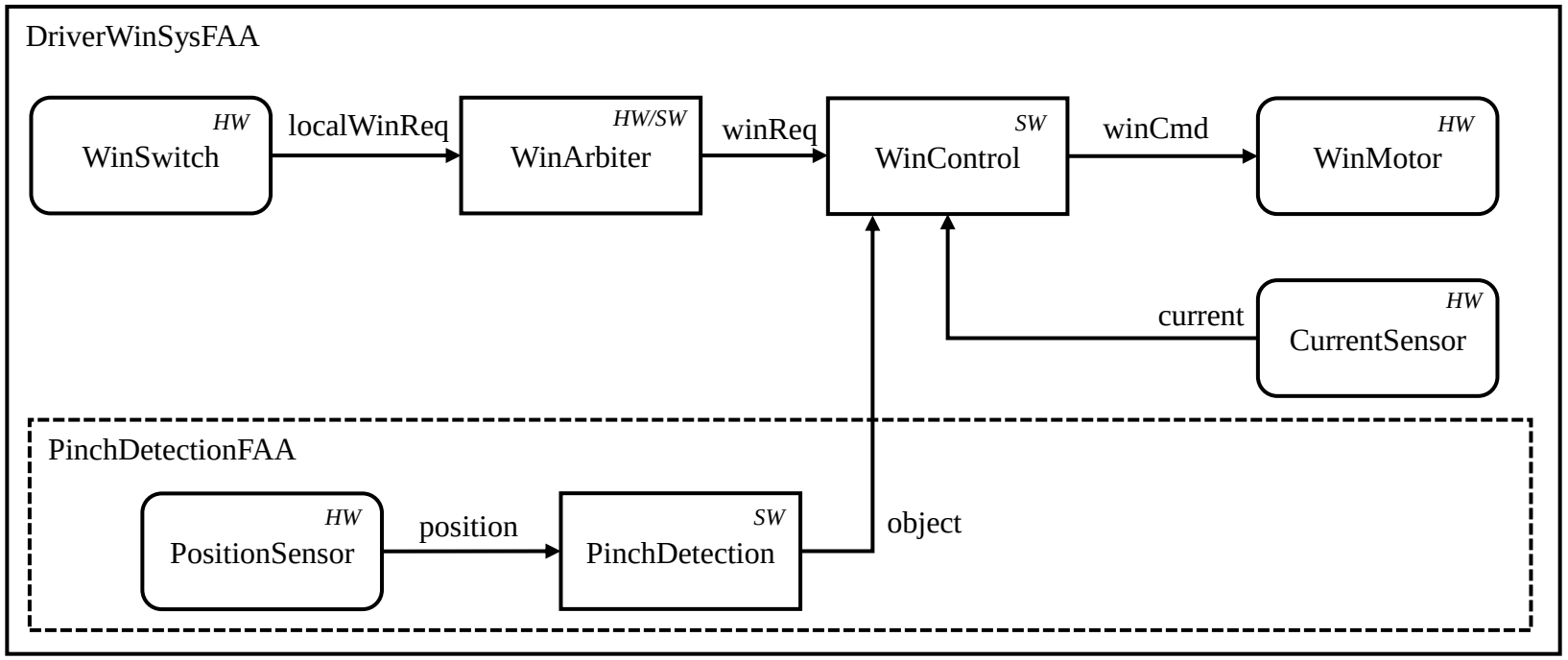


Capturing Variability

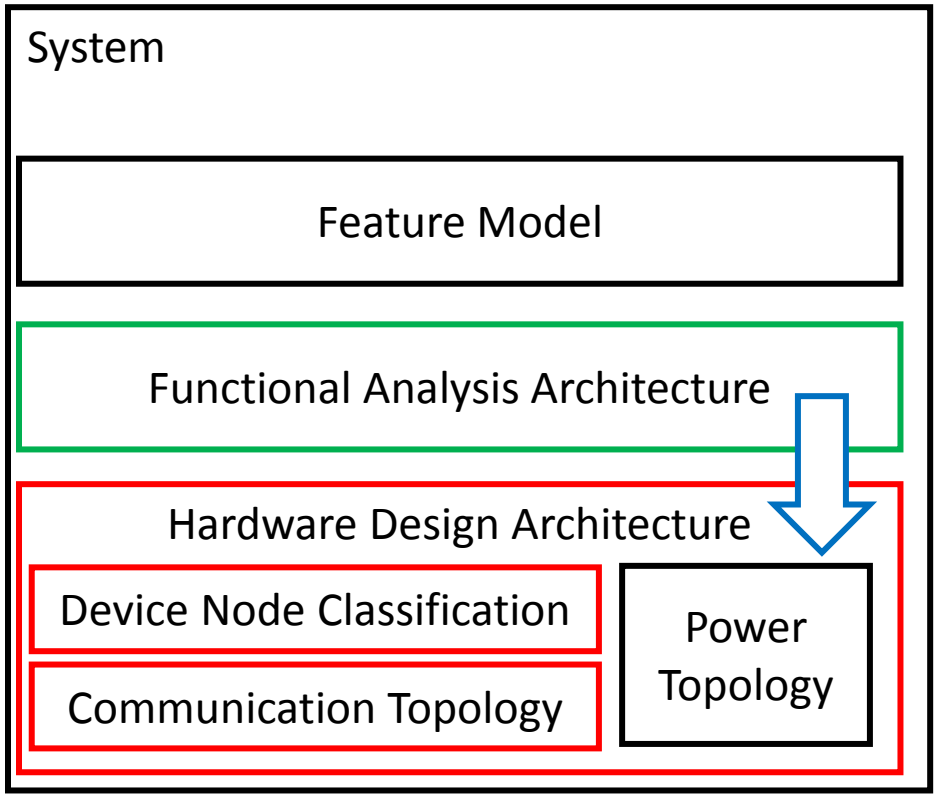


Emily's Power Window Example





Capturing Latency



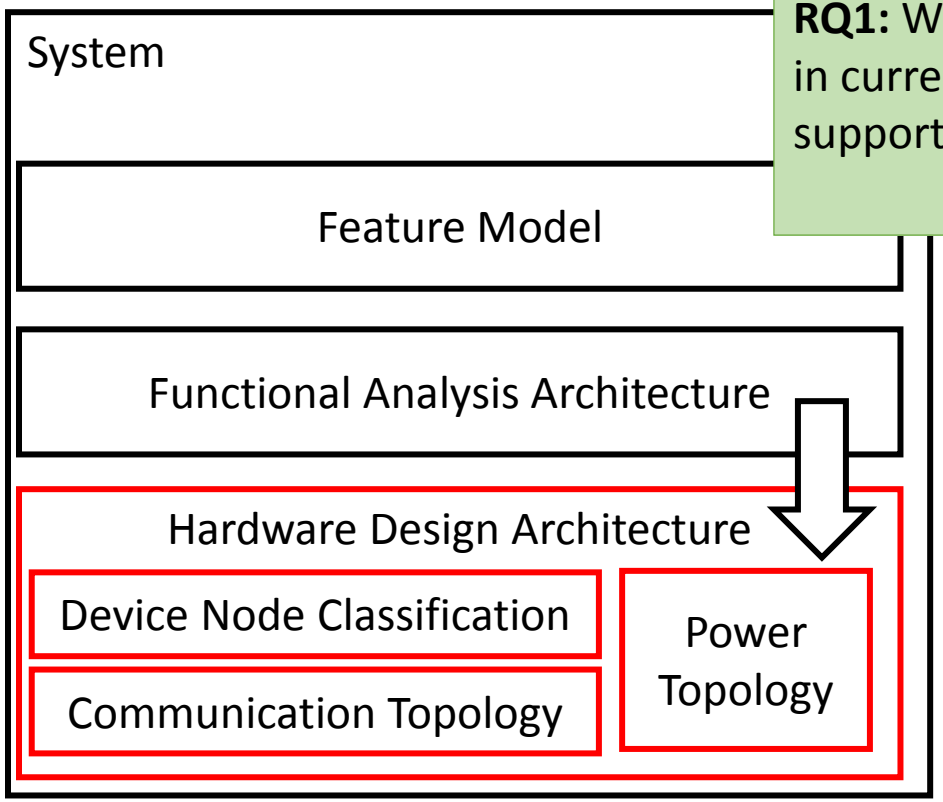
- Form timing chains using functions and function connectors
- Assign latencies to functional devices and analysis functions
- Assign message sizes to function connectors

- Deployed to smart node affects function latency
- Deployed to hardware connector affects function connector latency

- Assign speed factors to smart nodes
- Different communication connectors have different transfer rates.

Capturing Mass, Parts Cost, and Warranty Parts Cost

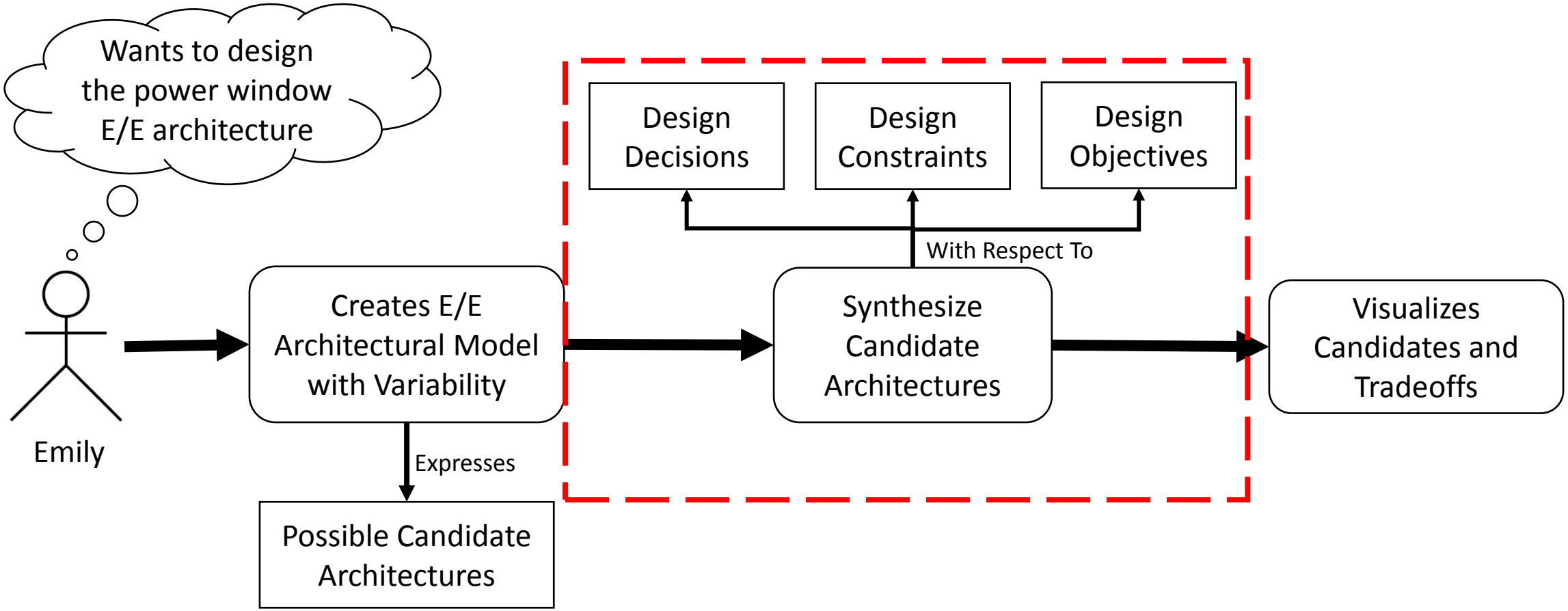
RQ1: What aspects of our reference model are unique and not found in current meta-models for E/E architecture? Or are found but not supported by reasoning?



- Assign mass to device nodes.

Cost
- Multiplicative factor for unit length cost for each connector type.
- Assign cost to device nodes.

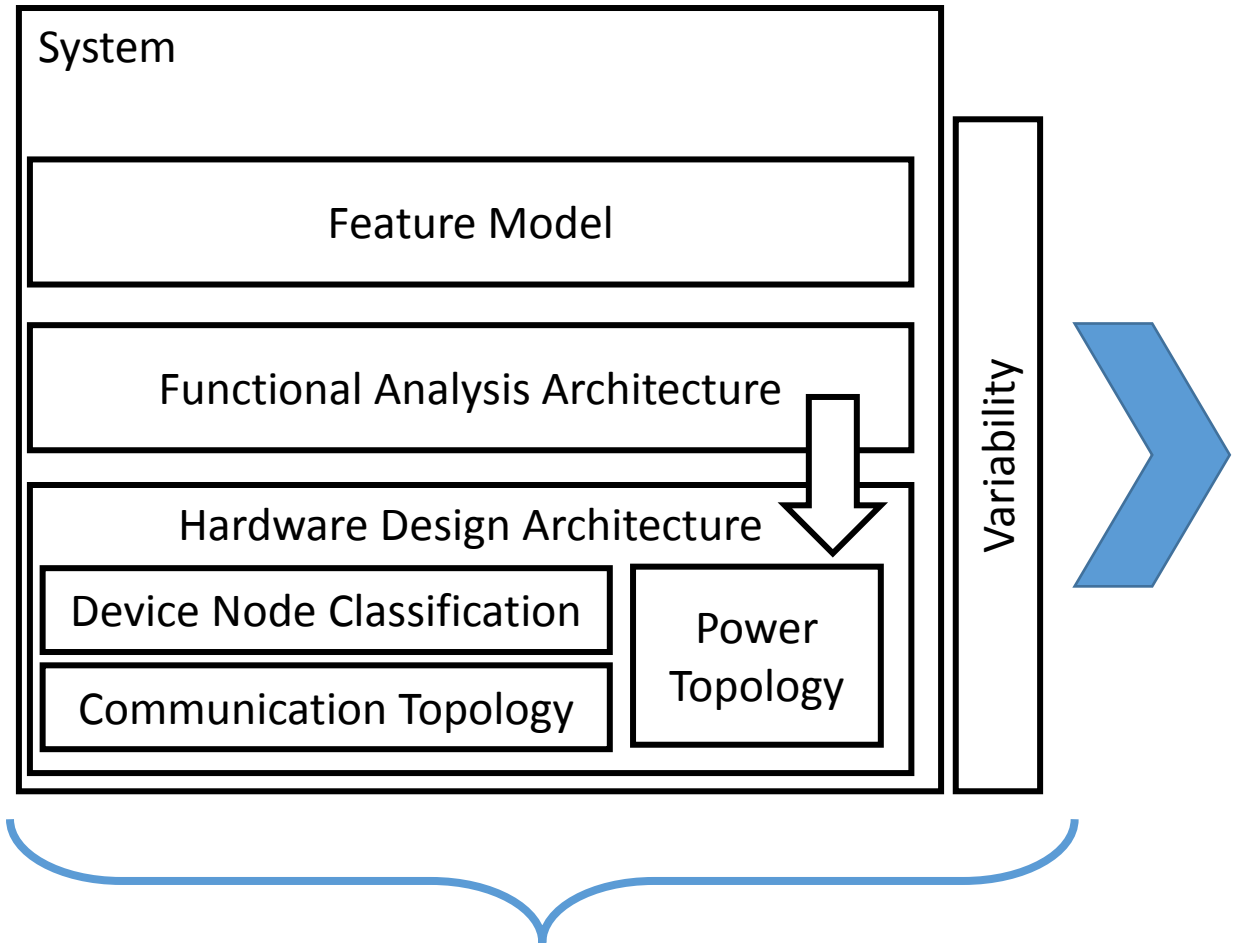
Warranty Cost
- Assign replacement cost to device nodes.
- Assign failure rate in PPM


Wants to design the power window E/E architecture
Emily
Creates E/E Architectural Model with Variability
Expresses
Possible Candidate Architectures
Design Decisions
Design Constraints
Design Objectives
With Respect To
Synthesize Candidate Architectures
Visualizes Candidates and Tradeoffs

Some Example Design Exploration Scenarios

1. Emily would like to investigate the possibility of adding a dedicated ECU to each door (we call the door module). Precisely, she would like to find out if it is a cost effective solution while meeting the requirements for mass and latency.
2. Emily is tasked with designing the power window for a higher end car in which cost is irrelevant but mass should be **minimized**, she would like to explore the possible designs. Additionally, since it's a high end car, all features should be included. Lastly, the end-to-end latency for pinch detection to react and reverse the motor should be less than 200 ms.
3. Emily would like to **minimize** the cost, regardless of the features to support an "economy class" vehicle her company is rolling out. Is there an optimal car design that does include all features?

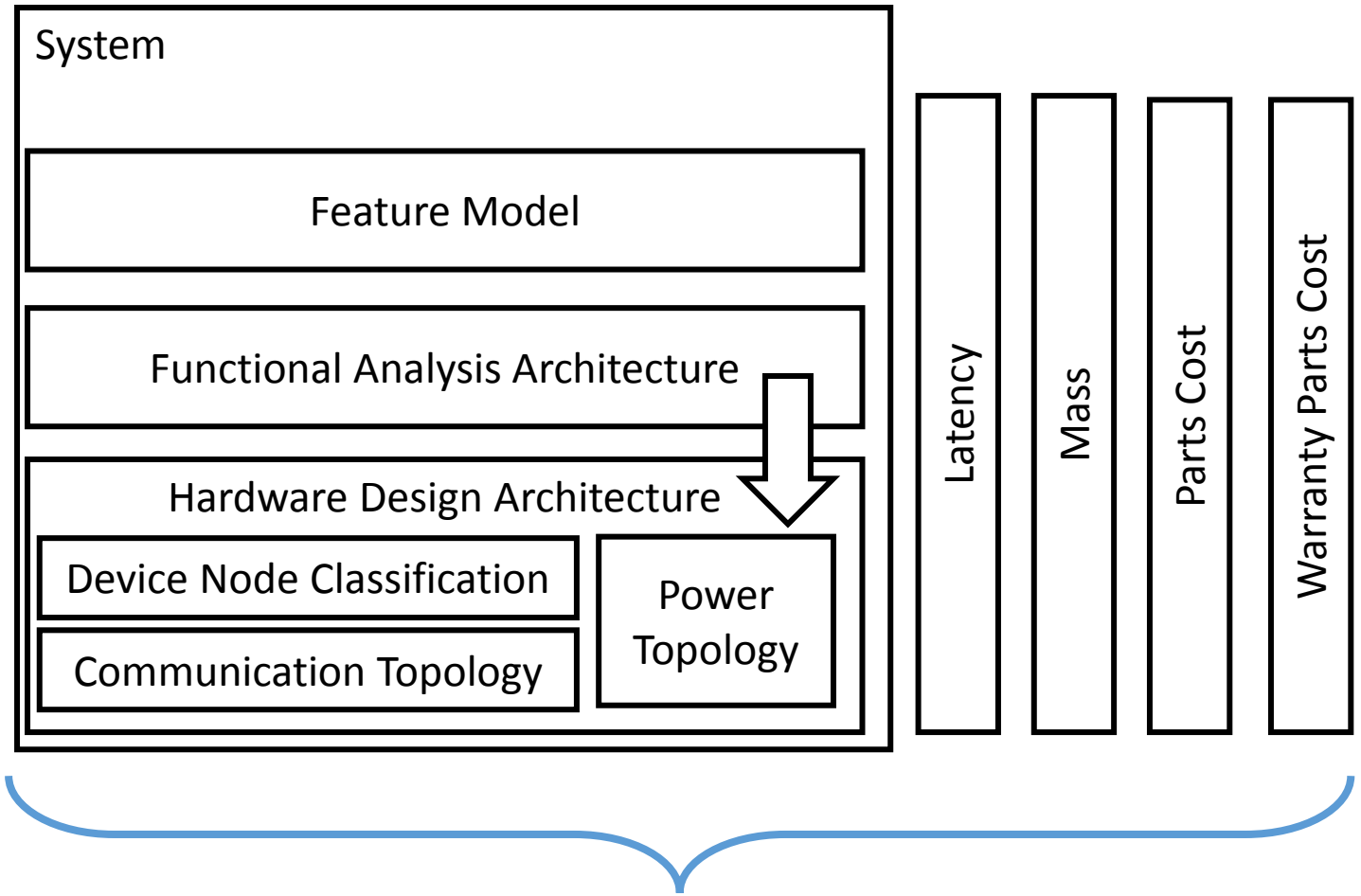
What Design Decisions Can We Make?



1. Feature “ExpressUp” is in the architecture.
2. “WinArbiter” function is implemented in hardware.
3. The “Switch” device node is smart
4. The “DoorModule” device node is present in the architecture
5. The “winCmd” function connector uses the “localDoorBus” bus connector to communicate.

….

What Design Constraints and Objectives Can We Have?



1. The end to end latency for the timing chain from the “WinSwitch” function to the “WinMotor” function is less than 500 ms.
2. Minimize the total mass of the system

…

Combine the system model with quality perspectives

Generalizing the Possible Specifications

Reference Model Concept	Template Design Decision
Features	Feature X is present in the architecture
Functions & Connectors	Function X is deployed to device node(s) Y(1) or Y(2) or ... Y(m) Function X is present in the architecture Function connector X is present in the architecture Function X is implemented in <hardware|software>
Device Nodes	Device node X is <smart|electric/electronic|power> Device node X is present in the architecture
Power Connectors	<Load|Device> power connector X is present in the architecture The <load|device> power connector X should be provided <to|from> device node(s) Y(1) or Y(2) or ... Y(m)
Communication Connectors	<Bus|Discrete|Analog> connector X is present in the architecture Function connector X does not use a hardware connector to communicate Function connector X uses connector(s) Y(1) or Y(2) or ... Y(m) to communicate Bus Connector X is of type <LIN|Low Speed CAN|High Speed CAN|FlexRay> The <bus|discrete|analog> connector X should have device node(s) Y(1) or Y(2) or ... Y(m) as endpoints

Quality Attribute	Template Design Constraint	ID
Timing (End-to-End Latency)	The end-to-end latency for timing chain X must be <less|greater> <than|than or equal to> Y	DC1
	The largest end-to-end latency difference for timing chains X(1), X(2), ..., X(n) must be <less|greater> <than|than or equal to> Y	DC2
Timing (Margin)	The margin between the timing chain X end-to-end latency and the requirement latency of Y must be <less|greater> <than|than or equal to> Z	DC3
	For all margins between the timing chain(s) X(1), X(2), ...X(n) end-to-end latency(s) and the requirement latency(s) Y(1), Y(2), ...Y(n) the <minimum|maximum> must be <less|greater> <than|than or equal to> than Z	DC4
Mass	The total mass of the architecture must be <less|greater> <than|than or equal to> Y	DC5
	The mass of X must be <less|greater> <than|than or equal to> Y	DC6
	The sum for mass of components X(1), X(2), ...X(n) must be <less|greater> <than|than or equal to> Y	DC7
Parts Cost	The total cost of the architecture must be <less|greater> <than|than or equal to> Y	DC8
	The cost of X must be <less|greater> <than|than or equal to> Y	DC9
	The sum for cost of components X(1), X(2), ...X(n) must be <less|greater> <than|than or equal to> Y	DC10
Warranty Parts Cost		DC11
		DC12
		DC13

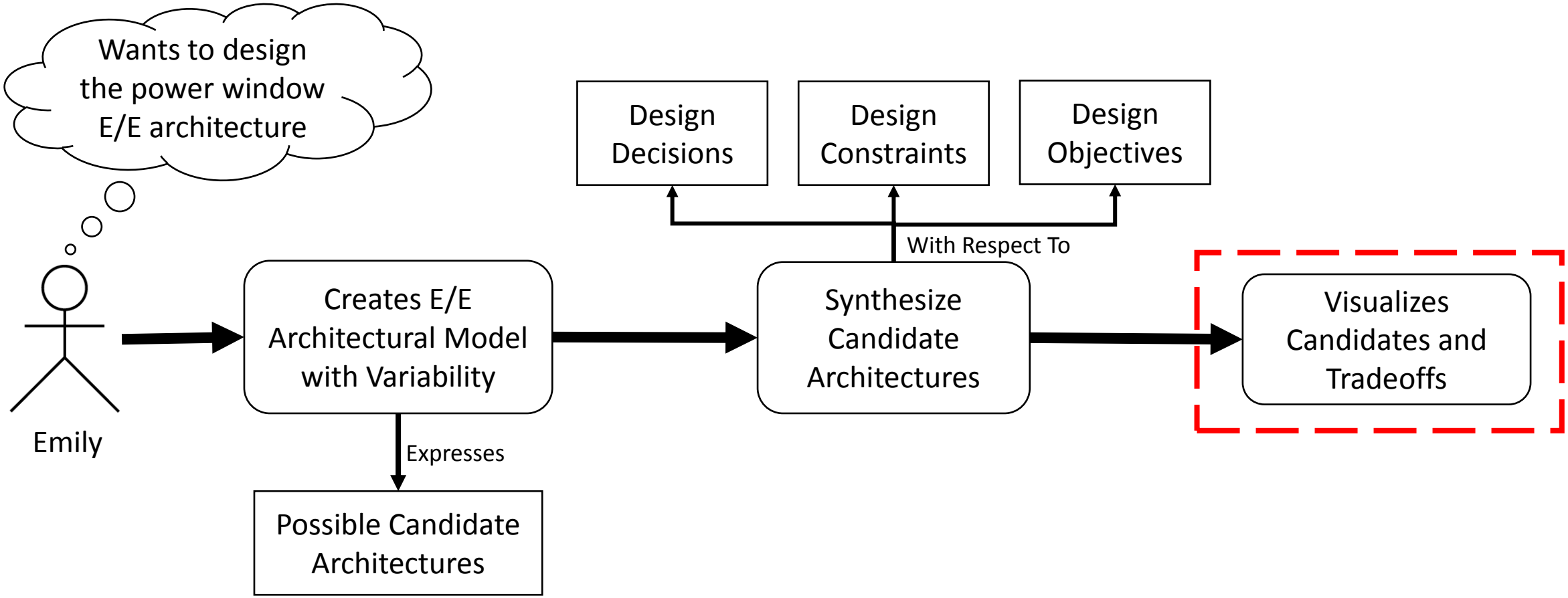
Quality Attribute	Template Design Objectives	ID
Timing (margins)	<Maximize|Minimize> the margin between the timing chain X end-to-end latency and the requirement latency of Y	DO1
	<Maximize|Minimize> the <smallest|largest> margin between the timing chain(s) X(1), X(2), ...X(n) end-to-end latency(s) and the requirement latency(s) Y(1), Y(2), ...Y(n)	DO2
Mass	<Maximize|Minimize> the total mass of the architecture	DO3
Parts Cost	<Maximize|Minimize> the total cost of the architecture	DO4
Warranty Parts Cost	<Maximize|Minimize> the total warranty of the architecture	DO5

Example

RQ2: Are there design exploration scenarios in which considering our reference model we can consider while others can not?

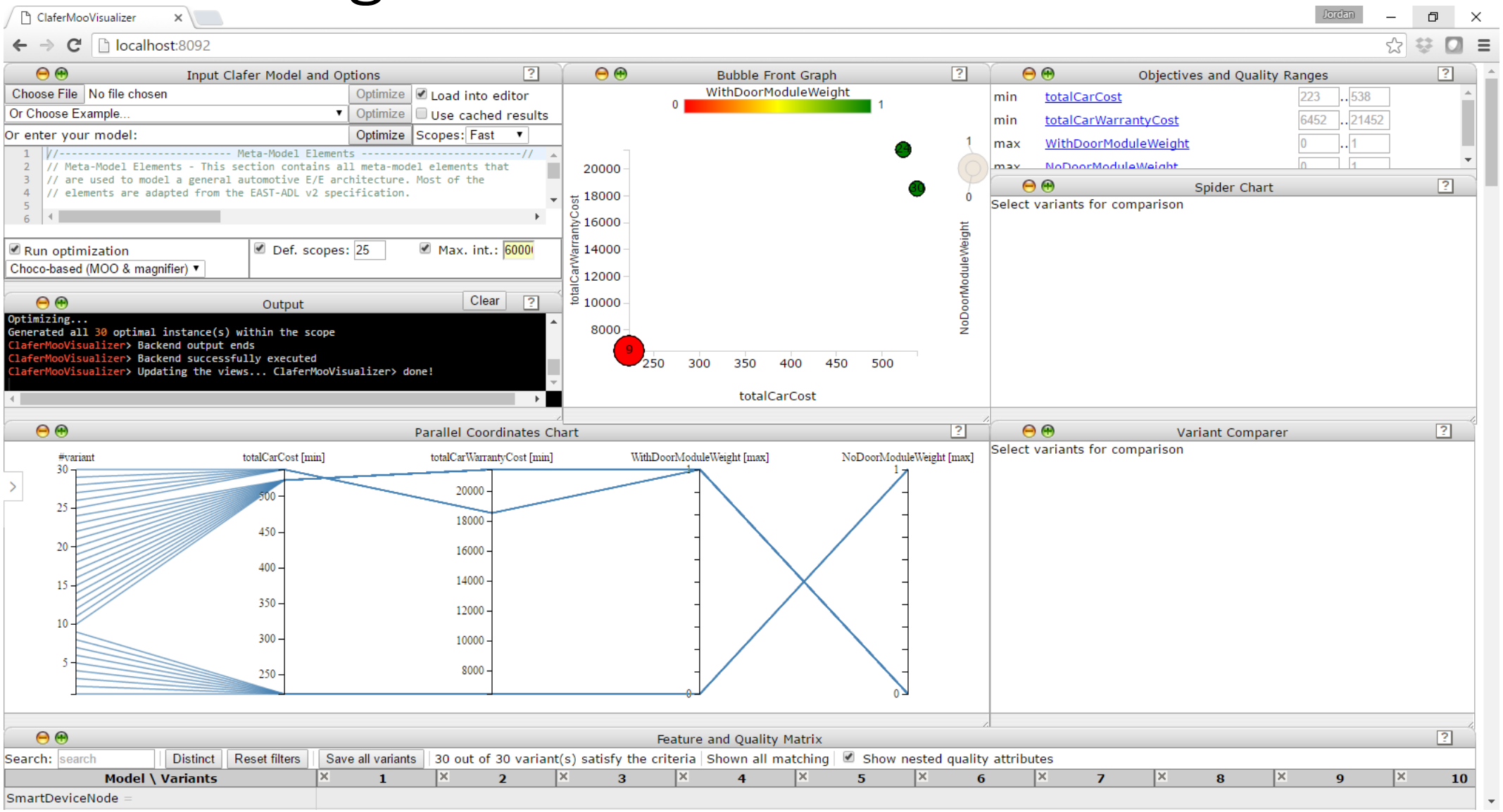
Emily is tasked with designing th
is irrelevant but mass should be minimized, she would like to explore the possible designs. Additionally, since its a high end car, all features should be included. Lastly, the end-to-end latency for pinch detection to react and reverse the motor should be less than 200 ms.

Feature “ExpressUp” is in the architecture AND The end-to-end latency for timing chain PinchDetection_TC must be less than 200 ms AND Minimize the total mass of the architecture


Wants to design the power window E/E architecture
Emily
Creates E/E Architectural Model with Variability
Expresses
Possible Candidate Architectures
Design Decisions
Design Constraints
Design Objectives
With Respect To
Synthesize Candidate Architectures
Visualizes Candidates and Tradeoffs

How Is This All Possible?

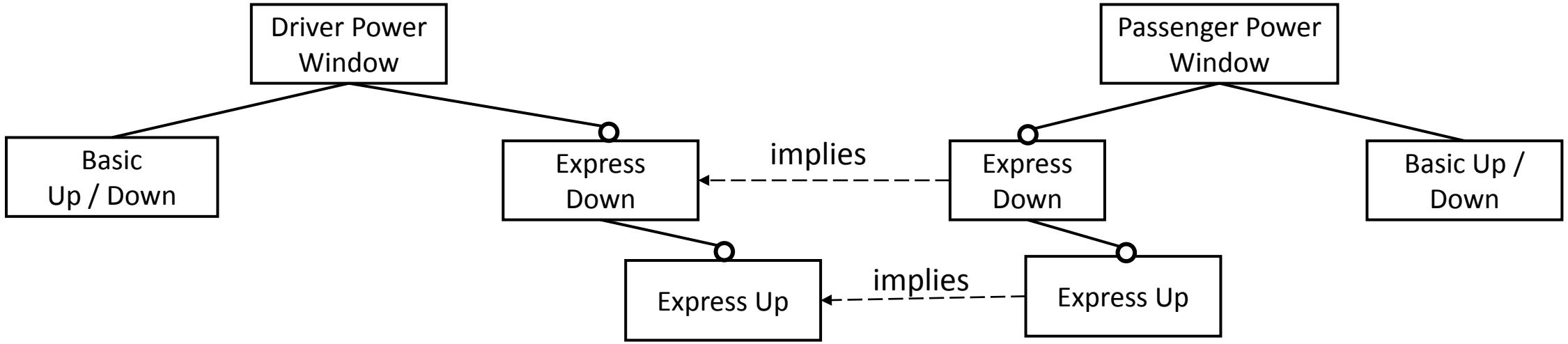
Visualizing Tradeoffs With Clafer Web Tools



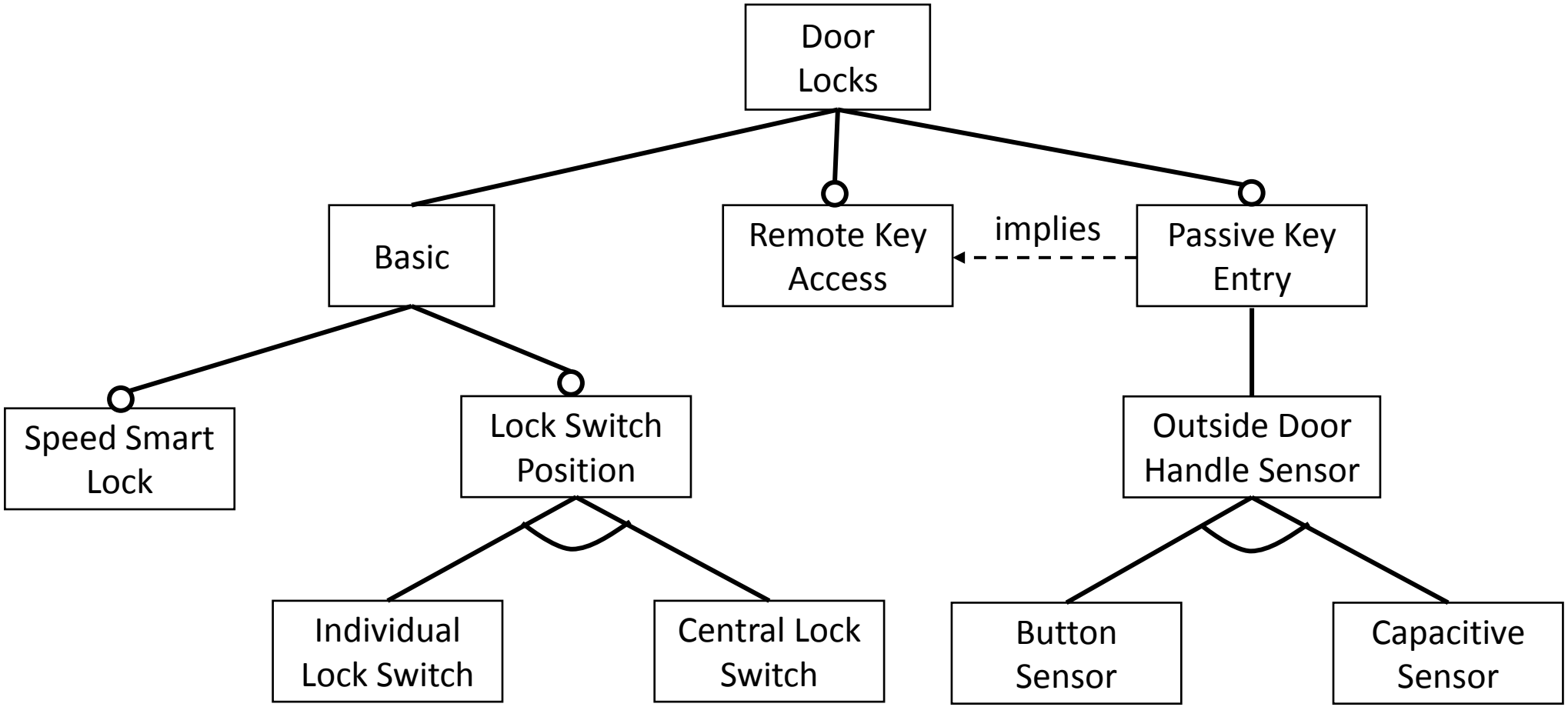
Case Studies

https://github.com/gsdlab/ClaferCaseStudies/tree/master/PlainClafer/Automotive/BodyDomain

Power Window Feature Model



Door Locks Feature Model



Model Sizes

	Single Door Power Window	Two Door Power Window	Central Door Locks
Features	3 (2)	6 (4)	7 (6)
Analysis Functions	3 (1)	6 (2)	3 (2)
Functional Devices	4 (1)	9 (2)	33 (15)
Deployment Configurations	64	4096	96
Function Connectors	6 (2)	7 (4)	33 (18)
Device Nodes	6 (2)	10 (3)	21 (14)
Discrete/Analog Connectors	13 (13)	18 (18)	34 (30)
Bus Connectors	1 (1)	2 (1)	2 (1)
Number of Variants	32 thousand	> 959 million	~ 2 thousand

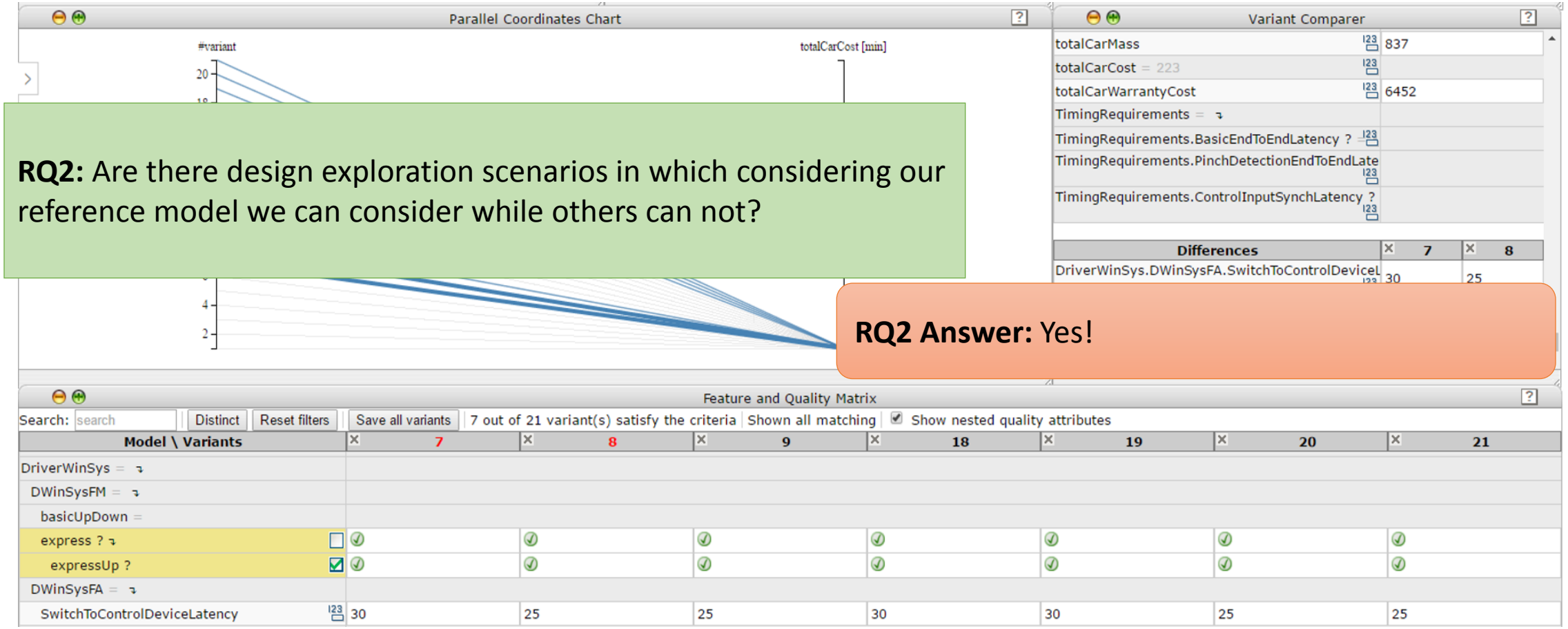
How Does Our Approach Compare?

RQ1: What aspects of our reference model are unique and not found in current meta-models for E/E architecture? Or are found but not supported by reasoning?

RQ1 Answer: Features, variability at all layers, function implementation, discrete/analog connectors, and power topology

Scenario	Tools			
	ArcheOpterix	*AF3*	*OSATE*	*AAOL*
express up feature		ally check satisfiable deployments.	rations.	
Dumb vs. Smart	No. Can't model switches and motors, discrete/analog connectors. No support for warranty cost.	No. Can't model discrete/analog connectors. No support for warranty cost.	No. No optimization support. No support for warranty cost. However, non-smart devices can be modeled.	No. Can't model discrete/analog connectors. No support for warranty cost.
High-end car	No. Can't constrain the latency of specific function combinations.	No. No optimization or modeling support for mass.	No. No optimization support, however detailed latency analysis can be done.	No. Can't constrain the latency of specific function combinations.
Economy car	No. Can't model variable features.	No. Can't model variable features.	No. Can't model variable features.	No. Can't model variable features.
Distributed vs. Centralized	No.	No. However, there is support to specify more than one deployment.	No. However, one could specify multiple system implementations to model two of the variants.	No. However, they have shown how they can model this for different qualities.

A Closer Look at the Economy Scenario



RQ2: Are there design exploration scenarios in which considering our reference model we can consider while others can not?

RQ2 Answer: Yes!

Looking at the Chocosolvers Performance

	Single Door Power Window			Two Door Power Window			Door Locks		
Design Spec.	*Solving Time (s)*	*Std. Dev. (s)*	*Number of Instances*	*Solving Time (s)*	*Std. Dev. (s)*	*Number of Instances*	*Solving Time (s)*	*Std. Dev. (s)*	*Number of Instances*
DD1	5.3	0.6	10	14.38	1.6	10	270.11	10.9	10
DD2	5.12	0.3	10	14.63	1.6	10	77.74	2.7	10
DD3	5.67	0.5	10	13.5	2.3	10	270.9	10.6	10
DD4	5.37	0.5	10	13.65	2.4	10	402.79	7.1	10
DD5	5.29	0.5	10	13.65	1.9	10	76.2	2.0	10
DD6	5.1	0.3	10	14.91	2.5	10	270.62	6.1	10
DD7	5.46	0.2	10	428.8	10.0	10	272.15	6.8	10
DD8	4.58	1.0	10	14.82	2.9	10	21.63	1.7	10
DD9	4.54	0.8	10	15.87	1.6	10	77.02	3.5	10
DD10	4.66	0.4	10	12.02	2.3	10	27.15	2.4	10
DD11	5.62	0.3	10	13.99	2.2	10	82.41	2.5	10
DD12	5.07	0.3	10	14.7	0.7	10	79.8	1.1	10
DD13	4.61	0.3	10	12.19	2.1	10	25.59	2.1	10
DD14	17.2	1.0	10	Timeout	-	-	75.15	1.3	10
DC1	4.37	0.4	10	13.69	2.1	10	76.44	1.5	10
DC2	4.88	0.4	10	13.8	2.5	10	78.03	2.9	10
DC3	4.62	0.3	10	14.18	2.0	10	76.81	1.6	10
DC4	5.16	0.4	10	14.61	1.6	10	76.95	1.8	10
DC5	4.7	0.4	10	151.5	6.4	0	80.38	2.0	
DC6	4.61	0.4	10	14.7	1.6	10	75.81	1.9	
DC8	4.74	0.3	10	Timeout	-	-	223.52	5.1	
DO1	Timeout	-	-	Timeout	-	-	Timeout	-	
DO2	Timeout	-	-	Timeout	-	-	Timeout	-	
DO3	30.96	1.5	9	Timeout	-	-	201.13	1.6	
DO3, DO4, DO5	37.87	1.5	9	Timeout	-	-	Timeout	-	

RQ4: Is it even feasible to ask the individual design decisions, constraints, and objectives shown earlier?

RQ4 Answer: The majority are feasible however, there are issues in when trying to find all optimal solutions.

DC8	4.74	0.3	10	Timeout	-	-	223.52	5.1	2
DO1	Timeout	-	-	Timeout	-	-	Timeout	-	-
DO2	Timeout	-	-	Timeout	-	-	Timeout	-	-
DO3	30.96	1.5	9	Timeout	-	-	201.13	1.6	1
DO3, DO4, DO5	37.87	1.5	9	Timeout	-	-	Timeout	-	-

Looking at the Chocosolvers Performance

RQ5: Is it feasible to ask the 6 design scenarios when considering the single and two door power window model?

RQ5 Answer: It is feasible when considering the single door power window, however not for the two door case.

Scenario					r Window	
					ev.	*Number of Instances*
1 - With ECU	216.23	7.8	33	90.07	3.4	0
1 - No ECU	27.12				-	-
2	5.13				1.0	10
3 - Smart	280.21				-	-
3 - Dumb	23.93	0.5	6	Timeout	-	-
4	25.56	0.6	3	Timeout	-	-
5	29.33	0.8	21	Timeout	-	-
6 - Centralized	-	-	-	Timeout	-	-
6 - Distributed	-	-	-	Timeout	-	-

Conclusion

- Presented a reference model for early design of E/E architectures.
- Showed how the reference model can be used to model a power window architecture that expresses millions of candidate designs.
- Highlighted where our approach surpasses current tooling.
- Room for future work…
  - Improve performance of analysis
    - Model simplification
    - Solvers
    - Surrogate models
  - Extending our reference model to accommodate fault tolerant architectures.
  - Refining the early design candidates to detailed designs.
  - ROI for design exploration tools.

Thanks for Listening!

Questions?